Enhancing VVC through CNN-based Post-Processing

Fan Zhang, Chen Feng and David R. Bull

Abstract

This paper presents a new Convolutional Neural Network (CNN) based post-processing approach for video compression, which is applied at the decoder to improve the reconstruction quality. This method has been integrated with the Versatile Video Coding Test Model (VTM) 4.01, and evaluated using the Random Access (RA) configuration using the Joint Video Exploration Team (JVET) Common Test Conditions (CTC). The results show coding gains on all tested sequences at various spatial resolutions over different quantisation parameter ranges, with average bit rate savings (based on Bjøntegaard Delta measurements) of 3.90% and 4.13%, when PSNR and VMAF are used as quality metrics respectively. The computational complexities of different CNN architecture variants have also been investigated.

Index Terms

Post-processing, VVC, video compression, CNN, machine learning

I. Introduction

The importance of video compression has increased significantly in recent years due to the tension between the large amount of high quality, more immersive video content consumed everyday and the available bandwidth provided by communication technologies [1]. To address this challenge, the Joint Video Exploration Team (JVET) is developing a new video coding standard, Versatile Video Coding (VVC) [2], which is due to be finalised in 2020. Compared to the current standard H.265/HEVC [3], VVC targets a 30%-50% performance improvement through integrating numerous advanced features and better support for high spatial resolutions, high dynamic range and 360° video formats. Alongside MPEG video standards, the Alliance for Open Media (AOMedia) was founded to develop open source and royalty-free media delivery solutions [4]. In 2018, AOMedia released its first video coding format AOMedia Video 1 (AV1) [5] primarily targeted at Internet streaming applications, which has also achieved consistent improvements over H.265/HEVC [6, 7].

Inspired by recent advances in machine learning, especially with deep convolutional neural networks (CNNs), several authors have reported significant enhancements to standard video coding algorithms [8, 9]. Numerous work has been published that integrates CNN models with intra prediction [10, 11], inter prediction [12, 13], transforms [14], quantisation [15], entropy coding [16], loop filters [17, 18], post-processing [19, 20] and format adaptation [21–24]. Although such CNN-based coding tools demonstrate potential for improving compression performance, few of them which have been adopted by the new MPEG standard VVC [2], mainly due to their relatively high computational complexity and the requirement for graphical processing hardware support.

In addition to conventional normative coding tools, post-processing is often employed at the video decoder to improve perceptual quality of the reconstructed video content and to reduce visual artefacts. Equally, such approaches can be integrated into the encoder as in-loop filters, allowing the frames with enhanced visual quality to also be used as a reference for encoding neighbouring frames when inter prediction is enabled. For the case of VVC, CNN-based post-processing and in-loop filtering approaches have been proposed [25–27], which offer bit rate savings for All Intra mode [28]. However the coding gains for the more commonly used Random Access (RA) configuration [28], are relatively low, and the trade-offs between network complexity and overall performance are often not justified.

In this paper, a novel CNN-based post-processing approach is presented for use with VVC in the RA configuration. The employed deep CNN architecture has been previously used for single image super-resolution and video format

*The authors acknowledge funding from EPSRC (EP/M000885/1) and the NVIDIA GPU Seeding Grants. The authors are with the Department of Electrical and Electronic Engineering, University of Bristol, Bristol, BS8 1UB, UK. {fan.zhang, cf18202, dave.bull}@bristol.ac.uk

adaptation [23, 24]. To achieve the optimised performance, this CNN was trained on a large video database containing VVC compressed content at various spatial resolutions, for different quantisation parameter (QP) groups. The proposed approach has been evaluated on the VVC Test Model (VTM) 4.01, with the results showing consistent improvement on all JVET test sequences for different QP ranges. The computational complexity of the proposed method has been further analysed for different CNN structure variants (residual block numbers) and correlated with overall coding gains.

The rest of the paper is organised as follows. Section II describes the proposed post-processing approach, the employed CNN architecture and the training strategy. Section III presents the evaluation results with discussion. Finally, a conclusion and future work are outlined in Section IV.

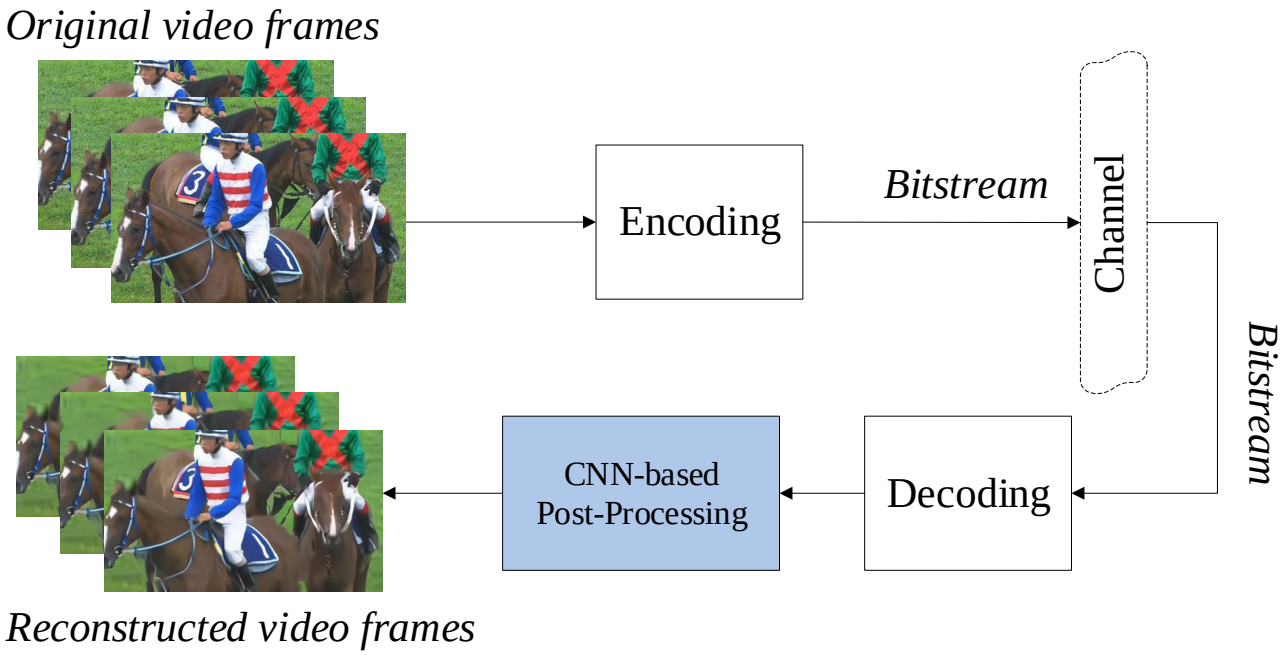


Fig. 1: Diagram of the proposed post-processing approach when it is integrated into a typical coding workflow.

II. Proposed Algorithm

The coding workflow with the proposed CNN-based post-processing approach is shown in Fig 1. This section provides a detailed description of the employed CNN architecture (Section II-A), the training content (Section II-B) and configuration (Section II-C), and the evaluation process (Section II-D).

A. Employed Network Architecture

Fig. 2 shows the CNN architecture used for post-processing VTM compressed content. The input to the network is a compressed RGB image block with a spatial resolution of 96×96 and a bit depth of 10, while the target is the corresponding original colour block in the same format. This CNN architecture has been employed in [23, 24] for spatial resolution and/or bit depth up-sampling, and was modified based on the generator (SRResNet) of SRGAN [29]. It contains $2N$+2 convolutional layers, all of which have 3×3 kernels, 64 feature maps and a stride value of 1, except the last convolutional layer (with 3 feature maps instead). Between the first and the last convolutional layers, there are N identical residual blocks, each of which contains two convolutional layers and a parametric ReLU activation function in between them. Skip connections are employed (i) between the input of each residual block and the output of the second convolutional layer of in the same residual block (ii) between the input of the first residual block and the output of the N^{th} residual block (iii) between the input of the CNN and the output of the last convolutional layer.

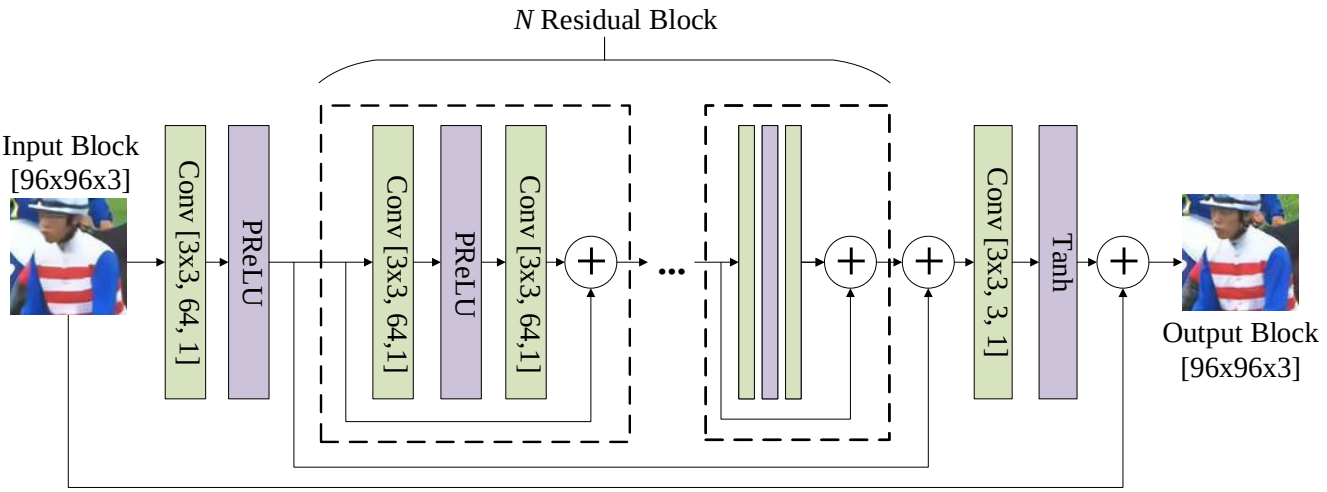


Fig. 2: The employed CNN architecture for post-processing.

B. Training Content

One hundred and eight source video sequences were used to train the employed CNN, selected from publicly available databases, including BVI-HFR [30], BVI-Texture [31], Harmonic 4K [32] and Netflix Chimera [33]. All were 10 bit, 3940×2160, 60 frames per second, YCbCr 4:2:0, raw video clips. These sequences were truncated to a length of 64 frames, and down-sampled to three lower resolutions, 1920×1080, 960×540 and 480×270, to increase content diversity. All of these 432 sequences (108 sources ×4 resolutions) were encoded by VVC VTM 4.01 [34] using the RA configuration of the JVET CTC. Other coding parameters include: five base quantisation parameter (QP) values 22, 27, 32, 27 and 42; Main10 profile; and a fixed intra period of 64. All the 432 reconstructed video sequences for each base QP value and their corresponding originals were randomly selected and segmented into 96×96 colour image blocks (converted to the RGB space). This results in approximately 500,000 image block pairs in total for five QPs. Here block rotation has been applied to achieve data augmentation and model generalisation.

C. Training Configuration

Based upon the training content generated, five different CNN models ($\mathrm{Model}_{\mathrm{QP22}}$, $\mathrm{Model}_{\mathrm{QP27}}$, $\mathrm{Model}_{\mathrm{QP32}}$, $\mathrm{Model}_{\mathrm{QP37}}$ and $\mathrm{Model}_{\mathrm{QP42}}$) were obtained for five QP groups. These are used for different base QP ranges in evaluation:

$$\mathrm{CNN} = \begin{cases} \mathrm{model}_{\mathrm{QP22}}, \text{ if } & \mathrm{QP}_{\mathrm{base}} \leq 24.5 \\ \mathrm{model}_{\mathrm{QP27}}, \text{ if } & 24.5 < \mathrm{QP}_{\mathrm{base}} \leq 29.5 \\ \mathrm{model}_{\mathrm{QP32}}, \text{ if } & 29.5 < \mathrm{QP}_{\mathrm{base}} \leq 34.5 \\ \mathrm{model}_{\mathrm{QP37}}, \text{ if } & 34.5 < \mathrm{QP}_{\mathrm{base}} \leq 39.5 \\ \mathrm{model}_{\mathrm{QP42}}, \text{ if } & \mathrm{QP}_{\mathrm{base}} > 39.5 \end{cases} \quad (1)$$

The CNN architecture was implemented and trained using the Tensorflow 1.8.0 framework, with a learning rate of 10^{-4} and weight decay of 0.1. Each CNN model has been trained for 200 epochs, using $\ell 1$ as lost function. The number of residual blocks (N) was the same (16) as in [23, 24, 29] for all five QPs. The relationship between N and post-processing performance was further investigated for QP42[1] by training various CNN models ($\mathrm{Model}_{\mathrm{QP42,N=4}}$, $\mathrm{Model}_{\mathrm{QP42,N=8}}$, $\mathrm{Model}_{\mathrm{QP42,N=12}}$ and $\mathrm{Model}_{\mathrm{QP42,N=16}}$) for different number of residual blocks used (N=4, 8, 12, and 16).

[1]Reconstructed frames for QP 42 contain more artefacts than for other tested QP values.

TABLE I: The compression performance of the proposed method and two JVET proposals [26, 27] benchmarked on original VTM.

Method	VTM-PP								N0254	O0079
Metric	PSNR				VMAF				PSNR	PSNR
QP Range	H-QPs		L-QPs		H-QPs		L-QPs		L-QPs	L-QPs
Class-Sequence	BD-rate	BD-PSNR	BD-rate	BD-PSNR	BD-rate	BD-VMAF	BD-rate	BD-VMAF	BD-rate	BD-rate
A1-Campfire	-3.27%	+0.06dB	-2.33%	+0.04dB	-5.57%	+1.00	-4.60%	+0.64	-1.87%	-1.75%
A1-FoodMarket4	-2.64%	+0.09dB	-2.01%	+0.05dB	-3.83%	+0.69	-2.95%	+0.32	-0.76%	-0.01%
A1-Tango2	-3.30%	+0.07dB	-2.89%	+0.03dB	-3.40%	+0.58	-2.97%	+0.22	-1.19%	-0.86%
A2-CatRobot1	-5.20%	+0.14dB	-5.19%	+0.07dB	-4.57%	+0.70	-4.39%	+0.31	-2.98%	-1.50%
A2-DaylightRoad2	-5.95%	+0.11dB	-7.09%	+0.07dB	-6.75%	+0.91	-7.15%	+0.47	-2.72%	-2.98%
A2-ParkRunning3	-0.77%	+0.03dB	-0.37%	+0.02dB	-2.26%	+0.44	-0.19%	+0.25	-0.94%	-0.55%
Class A	**-3.52%**	**+0.08dB**	**-3.32%**	**+0.05dB**	**-4.40%**	**+0.72**	**-3.71%**	**+0.37**	**-1.74%**	**-1.28%**
B-BQTerrace	-2.20%	+0.04dB	-0.97%	+0.01dB	-6.10%	+0.79	-1.10%	+0.41	-1.11%	-0.76%
B-BasketballDrive	-3.38%	+0.09dB	-3.08%	+0.06dB	-1.78%	+0.30	+2.65%	+0.14	-1.11%	-1.70%
B-Cactus	-3.37%	+0.09dB	-3.03%	+0.06dB	-5.08%	+0.75	-4.38%	+0.35	-1.24%	-1.68%
B-MarketPlace	-2.56%	+0.07dB	-2.30%	+0.06dB	-4.83%	+0.85	-4.00%	+0.45	-0.85%	-1.39%
B-RitualDance	-3.84%	+0.18dB	-3.46%	+0.16dB	-4.56%	+0.99	-2.57%	+0.49	-1.36%	-1.84%
Class B	**-3.07%**	**+0.09dB**	**-2.57%**	**+0.07dB**	**-4.40%**	**+0.72**	**-1.88%**	**+0.37**	**-1.13%**	**-1.47%**
C-BQMall	-5.62%	+0.23dB	-5.57%	+0.20dB	-4.73%	+0.83	-6.76%	+0.32	-1.40%	-3.90%
C-BasketballDrill	-3.91%	+0.16dB	-3.60%	+0.15dB	-3.83%	+0.75	-2.83%	+0.39	-1.04%	-2.44%
C-PartyScene	-4.07%	+0.16dB	-4.30%	+0.18dB	-5.86%	+0.92	-4.11%	+0.38	-1.46%	-3.91%
C-RaceHorses	-3.12%	+0.11dB	-2.09%	+0.08dB	-3.38%	+0.63	+1.23%	+0.32	-1.68%	-3.11%
Class C	**-4.18%**	**+0.16dB**	**-3.89%**	**+0.15dB**	**-4.45%**	**+0.78**	**-3.12%**	**+0.35**	**-1.39%**	**-3.34%**
D-BQSquare	-8.74%	+0.36dB	-9.64%	+0.35dB	-10.07%	+1.15	-11.64%	+0.56	-0.82%	-6.61%
D-BasketballPass	-6.14%	+0.27dB	-5.61%	+0.28dB	-5.41%	+1.26	-3.96%	+0.62	-1.73%	-4.59%
D-BlowingBubbles	-3.68%	+0.14dB	-3.75%	+0.15dB	-4.82%	+0.78	-3.76%	+0.35	-0.80%	-3.79%
D-RaceHorses	-4.81%	+0.20dB	-4.20%	+0.20dB	-5.24%	+1.06	-1.03%	+0.59	-2.19%	-4.90%
Class D	**-5.84%**	**+0.24dB**	**-5.80%**	**+0.25dB**	**-6.39%**	**+1.06**	**-5.10%**	**+0.53**	**-1.39%**	**-4.97%**
Overall	**-4.03%**	**+0.14dB**	**-3.76%**	**+0.12dB**	**-4.85%**	**+0.81**	**-3.40%**	**+0.40**	**-1.45%**	**-2.46%**
	BD-rate=-3.90%, BD-PSNR=0.13dB				BD-rate=-4.13%, BD-VMAF=+0.61					

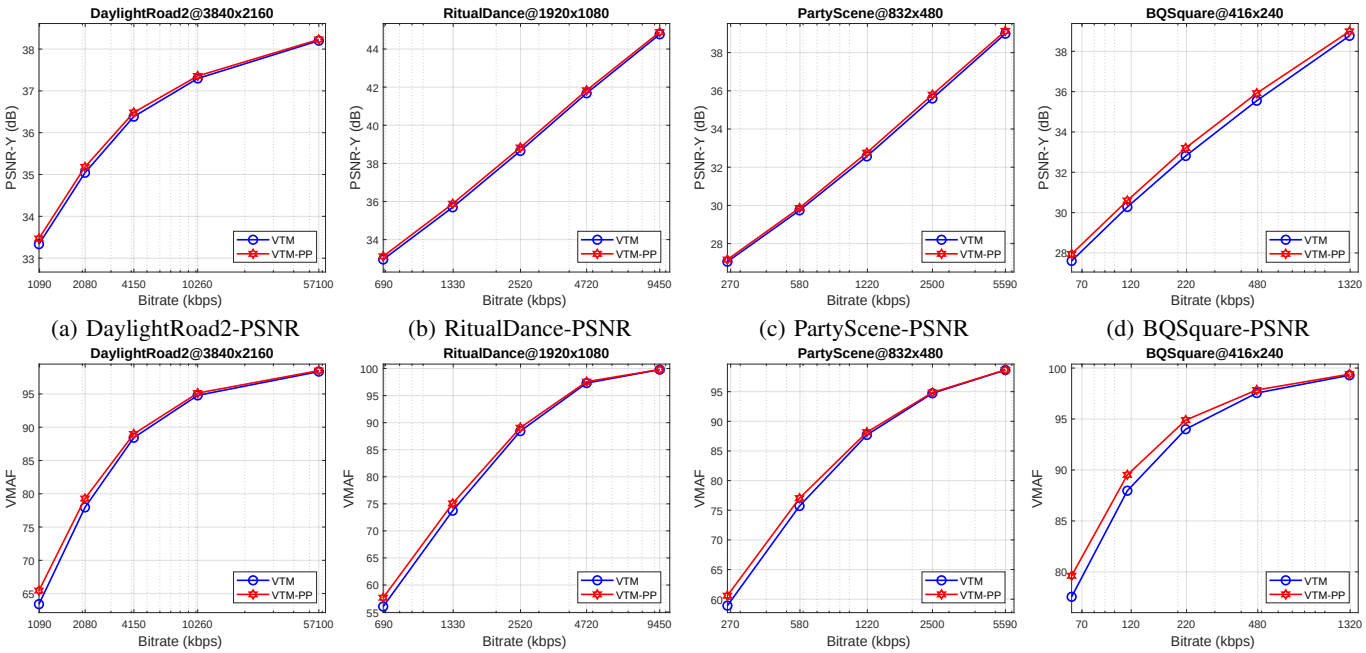
D. Evaluation for Large Video Frames

When the trained CNN models are employed for post-processing large video frames, each frame is segmented into 96×96 overlapping (with an overlap size of 4 pixels) image blocks with the same size as the CNN input. The final video frame is formed through aggregating all the CNN output image blocks in the same way.

III. Results and Discussion

The proposed method has been integrated with the VVC VTM 4.01 decoder and compared to the original VTM. Nineteen SDR (standard dynamic range) test sequences from JVET CTC video classes A1, A2, B, C and D were employed for evaluating the proposed method. None of these was included in the CNN training dataset mentioned in Section II-B.

The JVET CTC [28] RA configuration (Main10 profile) was employed for evaluation, using base QP values 22, 27, 32, 37 and 42. The first four QPs are recommended in JVET CTC, while QP 42 was used to extend the test bit rate range. The reconstructed quality was assessed using two video quality metrics, Peak Signal-to-Noise ratio (PSNR) and Video Multimethod Assessment Fusion (VMAF, version 0.6.1) [35]. The former is a simple but frequently used quality metric for image and video compression, while the latter combines multiple video features and quality metrics together through Support Vector Machine regression [36], offering improved correlation with subjective opinions for compressed content [37]. The rate quality performance was then evaluated through Bjøntegaard delta (BD) measurements for both low (22-37) and high QP (27-42) ranges.



(a) DaylightRoad2-PSNR (b) RitualDance-PSNR (c) PartyScene-PSNR (d) BQSquare-PSNR

(e) DaylightRoad2-VMAF (f) RitualDance-VMAF (g) PartyScene-VMAF (h) BQSquare-VMAF

Fig. 3: Rate-PSNR and Rate-VMAF curves for four selected test sequences. Here blue curves (VTM) stands for the original VTM 4.01, while red curves (VTM-PP) represent the proposed method.

In order to further benchmark the contribution of the proposed method, BD-rate results based on PSNR assessment for another two CNN-based post-processing/in-loop filtering approaches [26, 27] have also been presented. These two methods, denoted as JVET-N0254 and JVET-O0059, have been recently submitted to MPEG JVET meetings as VVC proposals.

As mentioned in Section II-C, the results using CNN models with different residual block numbers (N=4, 8, 12 and 16) were generated for all test sequences at QP 42. The computational complexity of the integrated decoder (VVC decoder plus post-processing) was also calculated for these versions. The training and evaluating processes were both executed on a shared cluster computer, in which each node contains 2.4GHz Intel CPUs, 138GB RAM and NVIDIA P100 graphic cards.

A. Compression Performance

The performance of the proposed method for all tested sequences is summarised in Table I, where original VTM 4.01 is used as a benchmark[2]. It is noted that the overall bit rate saving according to PSNR (for luma components only) is 3.76% for low QP range (22-37), which is higher than those for JVET-N0254 and JVET-O0059, and this improvement is evident for all tested video classes (resolutions). When VMAF is employed for quality assessment, the average BD-rates are similar to those for PSNR, at -4.85% for high QP range and -3.40% low QP range.

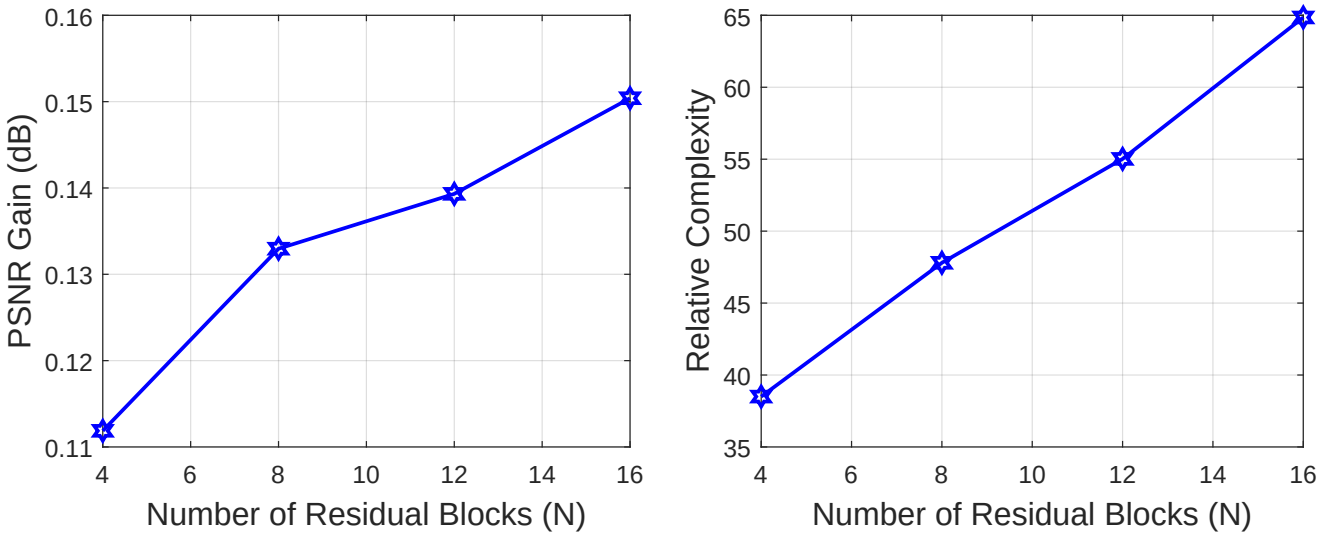
It can be also observed that, for the high QP range (27-42), the average coding gains are slightly higher than those for low QP range, and this difference is emphasised when VMAF is used for quality assessment. Rate-PSNR and rate-VMAF curves for four selected test sequences at various spatial resolutions, *DaylightRoad2, RitualDance, PartyScene and BQMall.*, are plotted in Fig. 3.

The coding gains can also be demonstrated by comparing the visual quality of the reconstructed frames generated by the original VTM and the proposed method. Fig. 5 shows example blocks of reconstructed frames for test sequences, *CatRobot1* and *BQMall*, with and without post-processing (when QP equals 42). It can be obserbed that the reconstructed frames after CNN post-processing exhibit fewer blocking artefacts and higher subjective quality compared to those generated by the original VTM decoder.

[2]Note that the results for JVET-N0254 and JVET-O0059 are benchmarked to the original VTM 4.0 and VTM 5.0 respectively.

B. Complexity Analysis

The relationship between the relative computational complexity (benchmarked to the original VTM 4.01 decoder) of the proposed method (VVC decoder plus post-processing) with different numbers of residual blocks, and the average PSNR gains for QP 42, on all 19 test sequences are illustrated by Fig. 4. It can be seen that when the number of residual blocks (N) employed decreases, the average PSNR gain over the original VTM reduces, as does the relative complexity.Both relationships (N versus PSNR Gain and N versus relative complexity) are approximately linear.



(a) (b)

Fig. 4: (a) PSNR gains for different number of residual blocks. (b) Relative complexity for different number of residual blocks.

IV. Conclusions

In this paper, a new CNN-based post-processing algorithm has been presented for use with the emerging Versatile Video Coding standard. Evaluation based on the VVC VTM 4.01 reference software, the proposed method shows enhanced reconstruction quality and coding gains compared against the original VTM and also against the start-of-the-art machine learning approaches. Future work will focus on integration of this method for in-loop filtering and on obtaining further improvements using perceptual quality metrics to train the CNN models.

V. References

[1] D. R. Bull, *Communicating pictures: a course in image and Video Coding*. Academic Press, 2014.

[2] B. Bross, J. Chen, S. Liu, and Y.-K. Wang, "Versatile video coding (draft 7)," in *the JVET meeting*, no. JVET-P2001. ITU-T and ISO/IEC, 2019.

[3] ITU-T Rec. H.265, *High efficiency video coding*, ITU-T Std., 2015.

[4] "Alliance for Open Media." [Online]. Available: https://aomedia.org/

[5] AOM. (2019) AOMedia Video 1 (AV1). [Online]. Available: https://github.com/AOMediaCodec

[6] A. V. Katsenou, F. Zhang, M. Afonso, and D. R. Bull, "A subjective comparison of AV1 and HEVC for adaptive video streaming," in *Proc. IEEE Int Conf. on Image Processing*, 2019.

[7] A. S. Dias, S. Blasi, F. Rivera, E. Izquierdo, and M. Mrak, "An overview of recent video coding developments in MPEG and AOMedia," in *International Broadcasting Convention (IBC)*, 2018.

[8] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, "Deep learning-based video coding: A review and a case study," *arXiv preprint arXiv:1904.12462*, 2019.

[9] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wanga, "Image and video compression with neural networks: A review," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[10] C.-H. Yeh, Z.-T. Zhang, M.-J. Chen, and C.-Y. Lin, "HEVC intra frame coding based on convolutional neural network," *IEEE Access*, vol. 6, pp. 50 087–50 095, 2018.

[11] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully connected network-based intra prediction for image coding," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3236–3247, 2018.

[12] Z. Zhao, S. Wang, S. Wang, X. Zhang, S. Ma, and J. Yang, "Enhanced bi-prediction with convolutional neural network for high efficiency video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[13] N. Yan, D. Liu, H. Li, B. Li, L. Li, and F. Wu, "Convolutional neural network-based fractional-pixel motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 840–853, 2018.

[14] D. Liu, H. Ma, Z. Xiong, and F. Wu, "CNN-based DCT-like transform for image compression," in *International Conference on Multimedia Modeling*. Springer, 2018, pp. 61–72.

[15] M. M. Alam, T. D. Nguyen, M. T. Hagan, and D. M. Chandler, "A perceptual quantization strategy for hevc based on a convolutional neural network trained on natural images," in *Applications of Digital Image Processing XXXVIII*, vol. 9599. International Society for Optics and Photonics, 2015, p. 959918.

[16] S. Puri, S. Lasserre, and P. Le Callet, "Cnn-based transform index prediction in multiple transforms framework to assist entropy coding," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 798–802.

[17] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, "Enhancing quality for hevc compressed videos," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[18] R. Yang, M. Xu, Z. Wang, and T. Li, "Multi-frame quality enhancement for compressed video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6664–6673.

[19] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in hevc intra coding," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 28–39.

[20] C. Li, L. Song, R. Xie, and W. Zhang, "Cnn based post-processing to improve hevc," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 4577–4580.

[21] X. Li, N. Oertel, A. Hutter, and A. Kaup, "Laplace distribution based Lagrangian rate distortion optimization for hybrid video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, pp. 193–205, 2008.

[22] M. Afonso, F. Zhang, and D. R. Bull, "Video compression based on spatio-temporal resolution adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 275–280, January 2019.

[23] F. Zhang, M. Afonso, and D. R. Bull, "Vistra2: Video coding using spatial resolution and effective bit depth adaptation," *arXiv preprint arXiv:1911.02833*, 2019.

[24] ——, "Enhanced video compression based on effective bit depth adaptation," in *Proc. IEEE Int Conf. on Image Processing*, 2019.

[25] M.-Z. Wang, S. Wan, H. Gong, and M.-Y. Ma, "Attention-based dual-scale CNN in-loop filter for Versatile Video Coding," *IEEE Access*, vol. 7, pp. 145 214–145 226, 2019.

[26] Y. Wang, Z. Chen, Y. Li, L. Zhao, S. Liu, and X. Li, "Ce13: Dense residual convolutional neural network based in-loop filter (ce13-2.2 and ce13-2.3)," in *the JVET meeting*, no. JVET-N0254. ITU-T, ISO/IEC, 2019.

[27] S. Wan, M.-Z. Wang, H. Gong, C.-Y. Zou, Y.-Z. Ma, J.-Y. Huo, Y.-F. Yu, and Y. Liu, "CE10: Integrated in-loop filter based on CNN (Tests 2.1, 2.2 and 2.3)," in *the JVET meeting*, no. JVET-O0079. ITU-T, ISO/IEC, 2019.

[28] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "JVET common test conditions and software reference

configurations for SDR video," in *the JVET meeting*, no. JVET-M1001. ITU-T and ISO/IEC, 2019.

[29] C. Ledig and *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 105–114.

[30] A. Mackin, F. Zhang, and D. R. Bull, "A study of high frame rate video formats," *IEEE Transactions on Multimedia*, vol. 21, no. 6, pp. 1499–1512, June 2019.

[31] M. A. Papadopoulos, F. Zhang, D. Agrafiotis, and D. R. Bull, "A video texture database for perceptual compression and quality assessment," in *Proc. IEEE Int Conf. on Image Processing*. IEEE, 2015.

[32] Harmonic, "Harmonic free 4K demo footage." [Online]. Available: https://www.harmonicinc.com/free-4k-demo-footage/#4k-clip-center

[33] I. Katsavounidis, "NETFLIX - "Chimera" video sequence details and scenes," November 2015. [Online]. Available: https://www.cdvl.org/documents/NETFLIX_Chimera_4096x2160_Download_Instructions.pdf

[34] J. Chen, Y. Ye, and S. Kim, "Algorithm description for Versatile Video Coding and test model 4 (VTM 4)," in *the JVET meeting*, no. JVET-M1002. ITU-T and ISO/IEC, 2019.

[35] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," *The Netflix Tech Blog*, 2016.

[36] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[37] F. Zhang, F. Mercer Moss, R. Baddeley, and D. R. Bull, "BVI-HD: A video quality database for HEVC compressed and texture synthesised content," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2620–2630, October 2018.

(a) CatRobots1 Original

(d) BQMall Original

(b) CatRobots1 VTM
(PSNR = 34.32dB)

(e) BQMall VTM
(PSNR = 31.90dB)

(c) CatRobots1 VTM-PP
PSNR = 34.51dB

(f) BQMall VTM-PP
PSNR = 32.15dB

Fig. 5: Example blocks of reconstructed frames generated by anchor VTM and the proposed methods for sequences, *CatRobot1* and *BQMall*, at QP 42, comparing to the uncompressed original. The PSNR (Y channel) values here are for the corresponding frames.